TA 1

Crear un componente Card, en ReactJS, que usando props despliegue la siguiente información:

- Título
- Descripción
- Persona asignada
- Fecha inicio
- Fecha de fin

Utilizar dicho componente para renderizar 3 Cards con diferente información.

El componente debe tener un estilo asociado, pero dicho estilo queda a decisión del alumno.

TA 2

Utilizando el componente creado anteriormente, modifiquelo para que en vez de mostrar la información utilizando props, use la prop children para renderizar la información solicitada.

TA 3

Crear un componente Input, en ReactJS, que permita al usuario ingresar información. Utilizar ese input para, en un párrafo aparte, mostrar la información que el usuario escribe.

TA 4

Crear una aplicación en ReactJS, que muestre en pantalla un contador. Dicho contador debe aumentar / disminuir su valor en función de 2 botones. Cada vez que se haga click en alguno de ellos, la aplicación debe volver a renderizarse para mostrar la información actualizada.

TA 5

Crear una aplicación en ReactJS donde al hacer click en un botón, un texto aparece o desaparece.

TA 6

Crear una aplicación en ReactJS, donde se le permite al usuario mediante un input ingresar una lista de tareas. Cada vez que se agrega una nueva tarea, el input debe quedar limpio, y la aplicación debe volver a renderizarse para mostrar la información actualizada.

TA 7

Utilizando como base la TA 6, agregar (de la manera que le parezca más amigable a nivel de UI), una manera de poder borrar una tarea.

TA 8

Utilizando como base la TA 7, agregar (de la manera que le parezca más amigable a nivel de UI), una manera de poder editar una tarea.

TA 9

Utilizando como base la TA 4, utilizar useEffect para que cada vez que el contador cambie, el título de su documento (lo que se muestra en la tab, cambie). Por defecto, debe ser Contador: 0

TA 10

Crear una aplicación en ReactJS, que utilizando useState y useEffect, llame a la siguiente API (https://jsonplaceholder.typicode.com/users) y muestre el nombre de usuario y el mail (de manera amigable) de la respuesta que obtiene.

Dicha llamada debe hacerse sólo una vez, cuando el componente se renderice.

TA 11

Crear una aplicación en ReactJS que muestre un temporizador que se incremente automáticamente cada segundo utilizando useEffect, y muestre dicho valor en pantalla.

Tome las precauciones necesarias para que, cuando el componente sea desmontado, dicho temporizador no quede corriendo.

TA 12

Crear un contexto en una aplicación React que guarde el nombre de un usuario. Usar los hooks necesarios para que dentro del App.jsx, dicha información sea cargada en el contexto, y dentro de un componente, se muestre el nombre de dicho usuario, obtenido desde el contexto (no se pueden usar props).

TA 13

Partiendo de la TA anterior, añadir la capacidad de actualizar el valor del nombre de usuario desde cualquier componente. Para eso, crear un componente input donde el usuario pueda escribir su nombre, y hacer que el valor del contexto se actualice en tiempo real.

TA 14

Crear un contexto que gestione un tema (oscuro o claro) en una aplicación ReactJS. Para ello, crear dos componentes: uno que cambie el tema y otro que cambie su estilo en función del tema actual.

Implementar dos estilos diferentes para los temas y cambia entre ellos cuando el contexto se actualice.

TA 15

Crear dos contextos diferentes: uno para gestionar el idioma de la aplicación y otro para gestionar el tema (oscuro/claro). Se tiene que proporcionar estos dos contextos a tu aplicación de manera que los componentes puedan consumir tanto el idioma como el tema.

Crear un componente que muestre el tema actual y otro que muestre el idioma actual.

TA 16

Objetivo: Crear una aplicación con navegación entre tres páginas: Home, About, y Contact.

Instrucciones:

- Instalar React Router usando npm install react-router-dom.
- Crear los componentes Home, About, y Contact.
- Usar BrowserRouter, Routes, y Route para definir las rutas.
- Implementa enlaces (Link) para navegar entre estas páginas.

TA 17

Objetivo: Mostrar contenido dinámico basado en parámetros de la URL.

Instrucciones:

- Crear una ruta para mostrar información de un producto usando un ID en la URL, por ejemplo: /product/:id.
- Usar useParams para acceder al parámetro id en la URL.
- Mostrar el contenido del producto dinámicamente basado en el ID.

TA 18

Objetivo: Crear rutas protegidas que solo permitan el acceso a usuarios autenticados.

Instrucciones:

- Crear una ruta protegida que redirija al Home si un usuario no cumple con cierta condición.

- Crear rutas que se encuentren protegidas por este mecanismo, así como públicas que no lo estén